

OpenAirInterface

Improvement of software quality

7th Nov. 2017

FUJITSU Limited

AKIYAMA Yuko (akiyama.yuko@jp.fujitsu.com)

Agenda

- Software Quality and Carrier Grade
- Approach to OAI quality improvement
- Extra

Software Quality and Carrier Grade



Carrier Grade

- 1 Extreme Stability**
No system/process/thread down. 24hrs/7days running.
- 2 Extreme Fault Tolerance**
Expect and get ready for unexpected things.
- 3 Extreme Recoverability**
Restart services immediately even after any fault occurs.

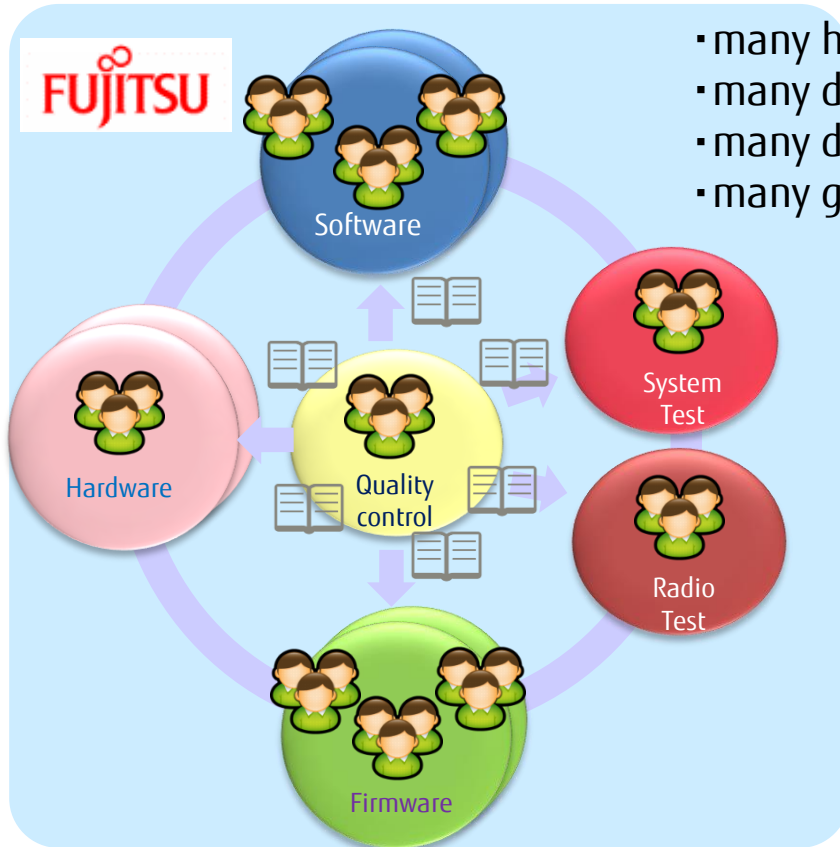
+

Well Tested

||

99.999% Availability

How FUJITSU secures Carrier Grade



- many human resources
- many department
- many development rules
- many gate of quality check



C-BBU (*)	
■	3000 UEs / 1 Cell
■	48 Cells
■	Major software update every 6 months



(*)Centralized Base Band Unit

Kano Model



Attractive Quality

Even if not implemented, it's OK. But once implemented, the customer becomes delighted.

One-dimensional Quality

The more you have, the more the customer gets satisfied.

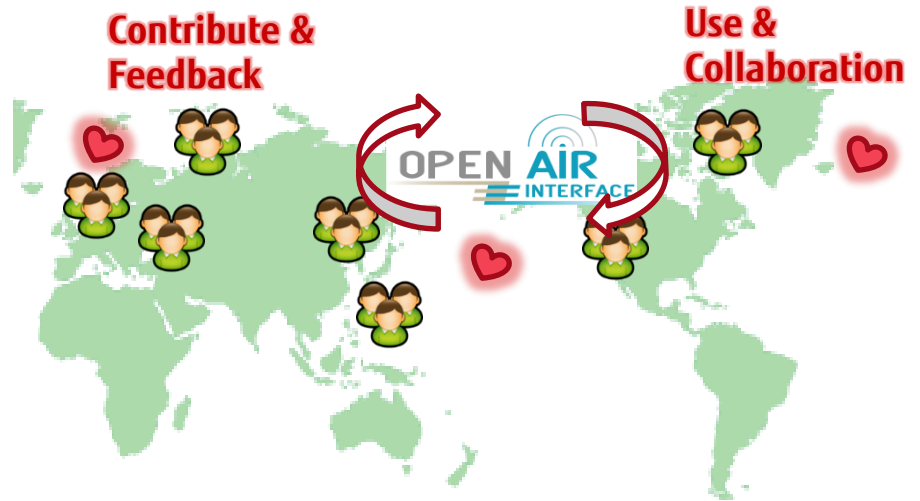
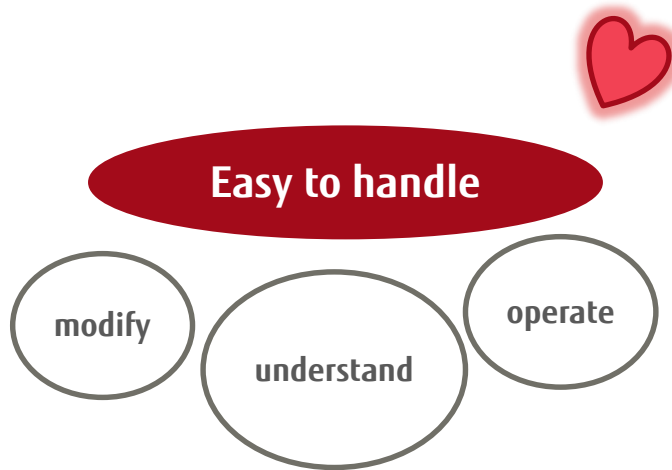
Must-be Quality

Even if fully implemented, the customer's reaction is neutral. But if not implemented, the customer gets really disappointed.

What **quality** do you want for OAI?

What is the state that software should be?

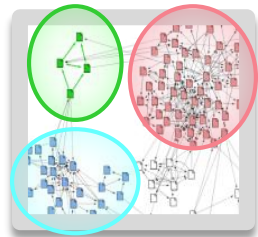
- The source code is always in an easy-to-understand state



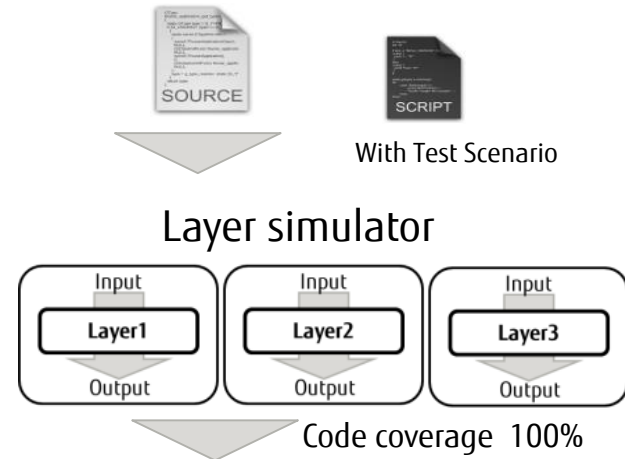
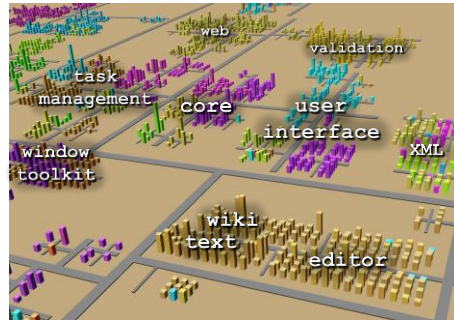
What should we do to secure this state?

Approach of Fujitsu on OAI

- SARF(*) : Utilization of software metrics tools
- Simulation testing



Graph clustering applied to dependency graph



Code Tested
by GDB, eclipse etc.

(*)Reference information at the end of presentation

SArF : Software Architecture Finder

■ Feature

- Software clustering based on dependency graph
- City map-like visualization of the detected clusters: SArF Map
- A city block corresponds to a detected software cluster

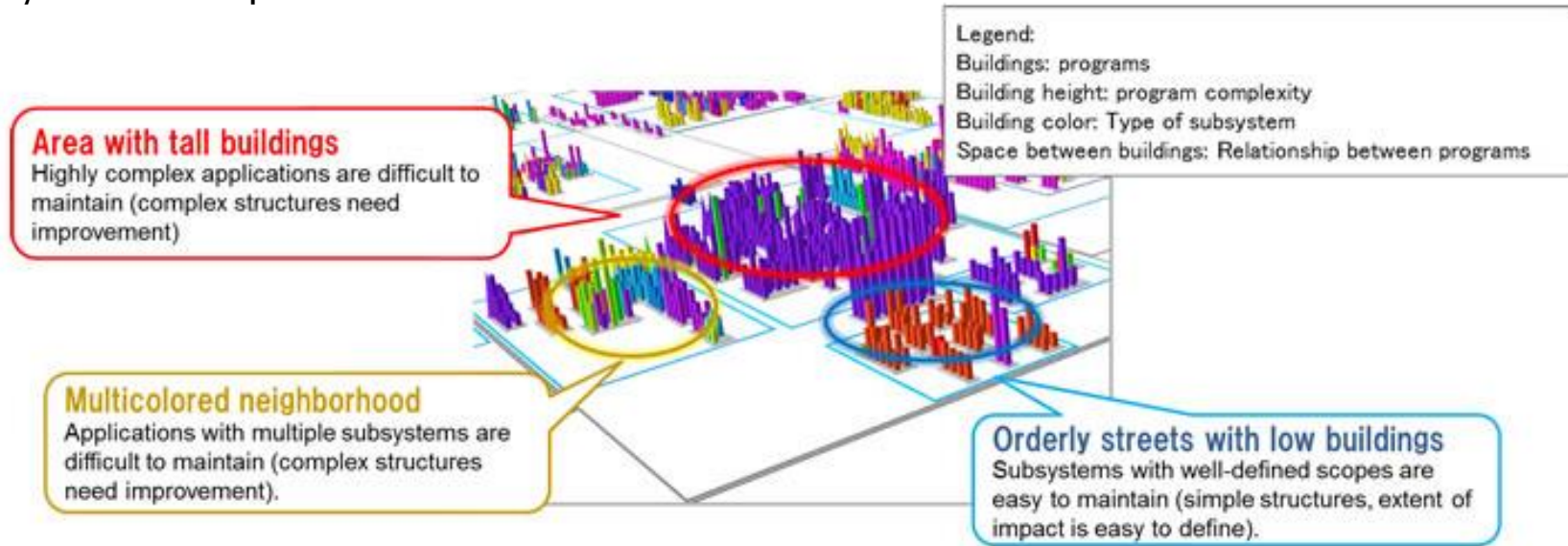
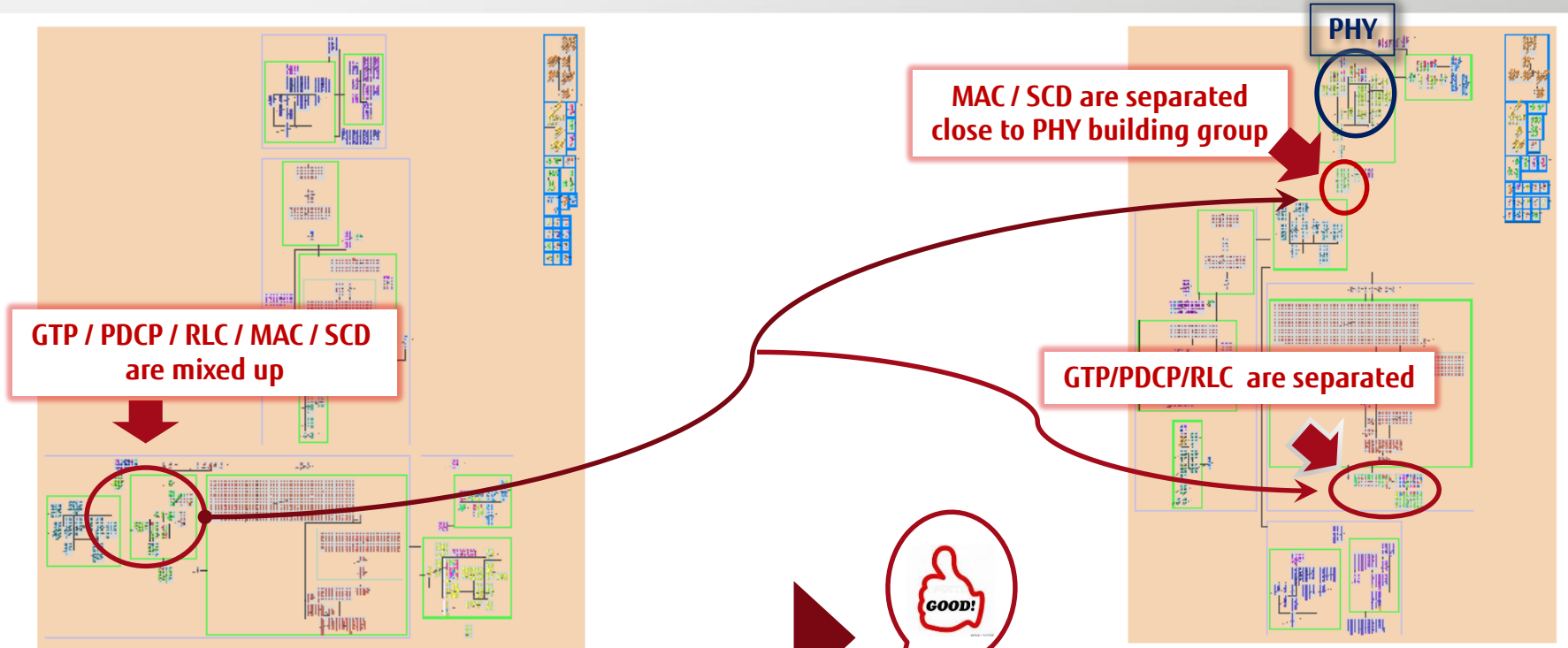


Figure 1: Sample image showing application structure visualized as a software map.

SArF Result OAI -Top view



Develop Branch_October, 23rd 2017

RU-RAU Split Branch_October, 23rd 2017

The architecture is getting better

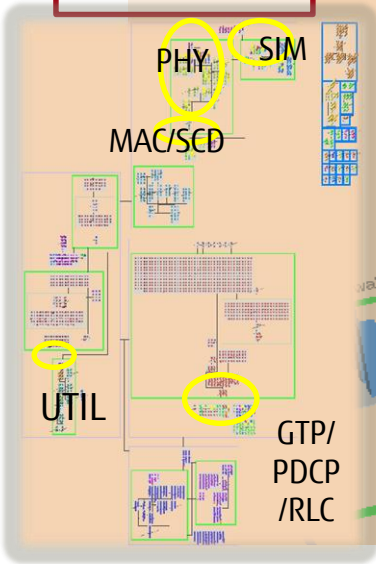
SArF Result OAI detail-3D

GTP/PDCP/RLC

MAC/SCD

Building height :
program complexity /LoC

TOP View



PHY

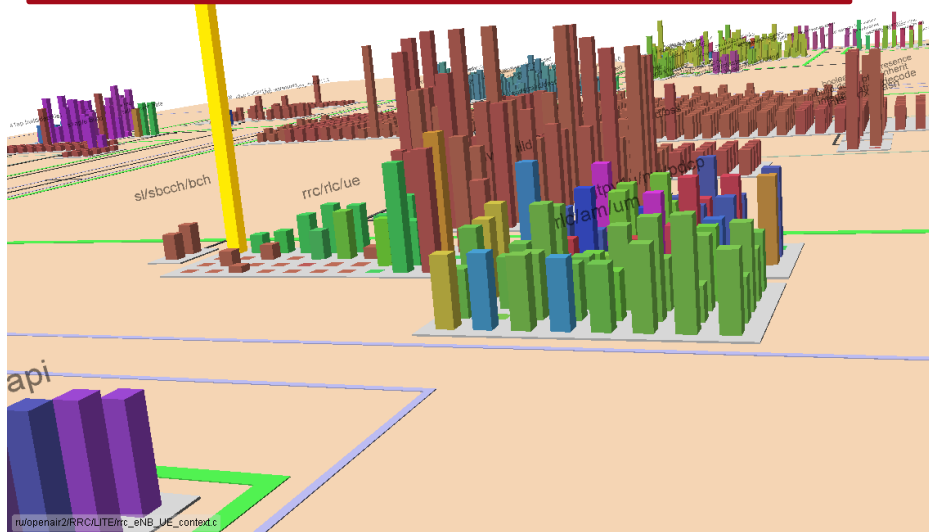


Yellow buildings refer to the same data, tables
→ This file is necessary to be refactored?

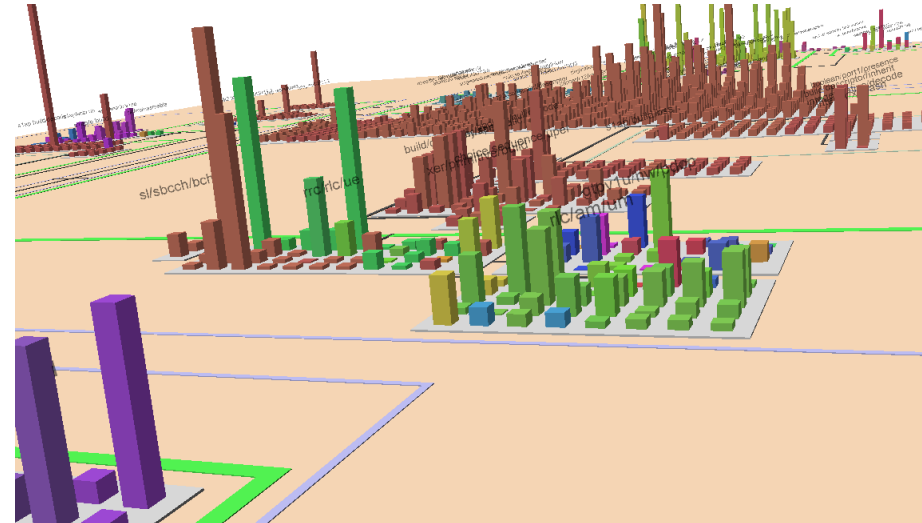
SArF Result OAI detail complexity vs LoC

RU-RAU Split Branch

Building height : program complexity /LoC



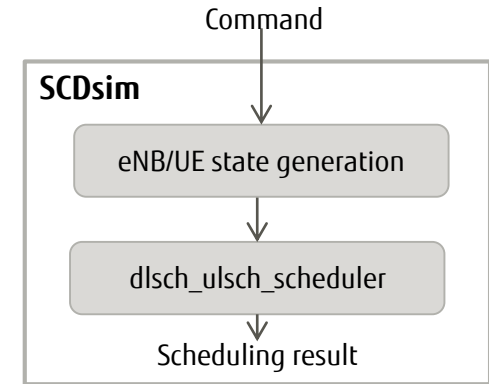
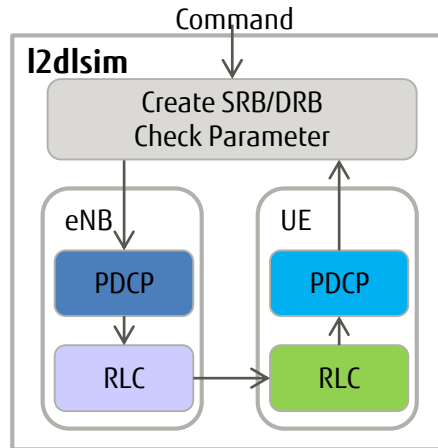
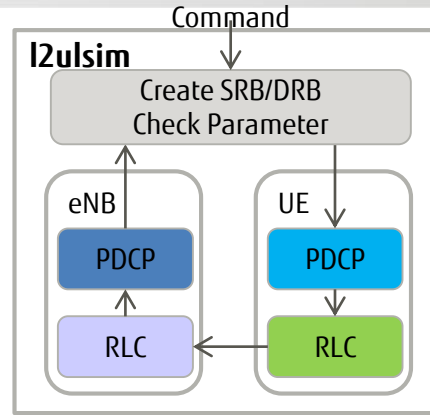
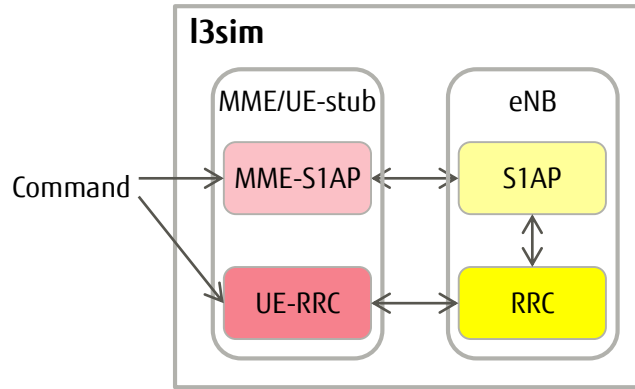
Building height : LoC



The building height can be set by various metrics which helps you check software visually.

Simulator(Layer simulator)

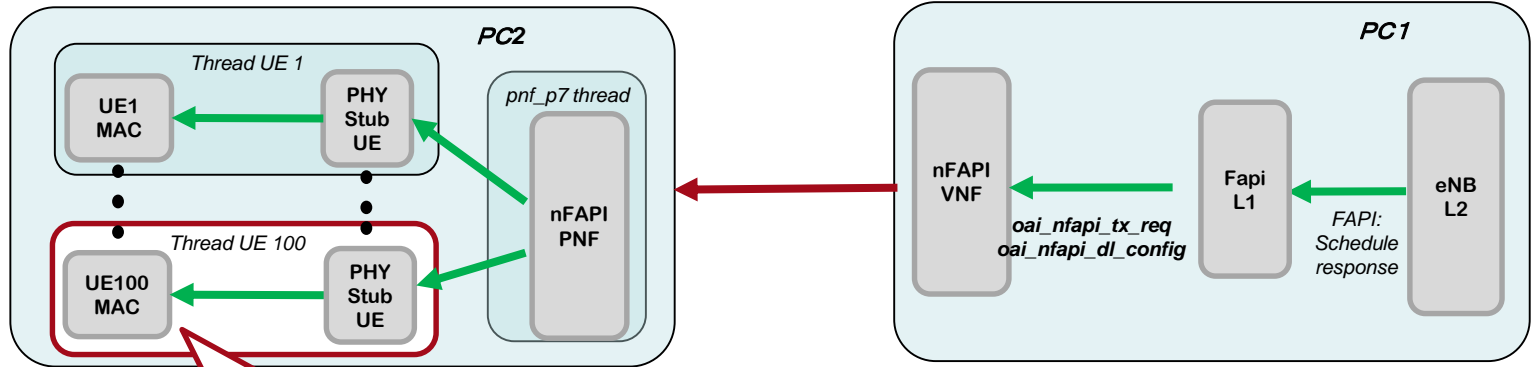
- Improvement of layer simulator



guarantee for each function properly

Simulator(OAI Sim)

- Improvement of OAI simulator



Extension to 100UEs

guarantee with OAI sim

What to do from now

- **Process**

Waterfall → Agile



Speed ↑

- **Quality**

Many gate of quality check

→ Continuous Integration, Auto Test, Simulation Test



Quality ↑

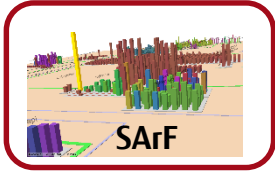
- **Development**

→ Commercially use as Private LTE



OAI users ↑

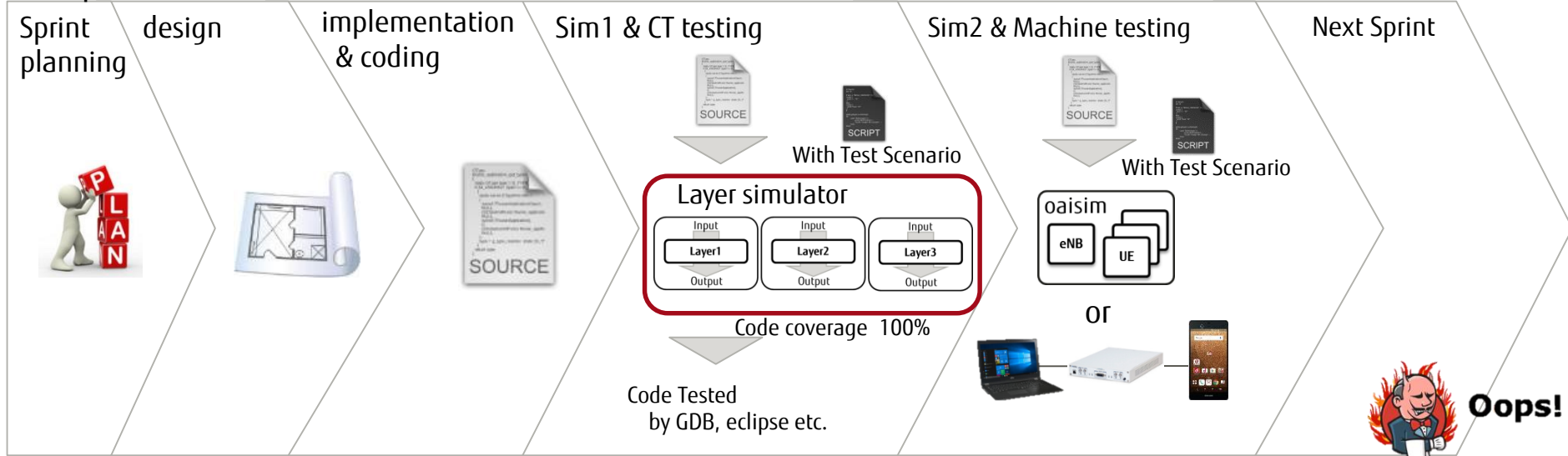
Fujitsu's software develop flow



<https://gitlab.eurecom.fr/>



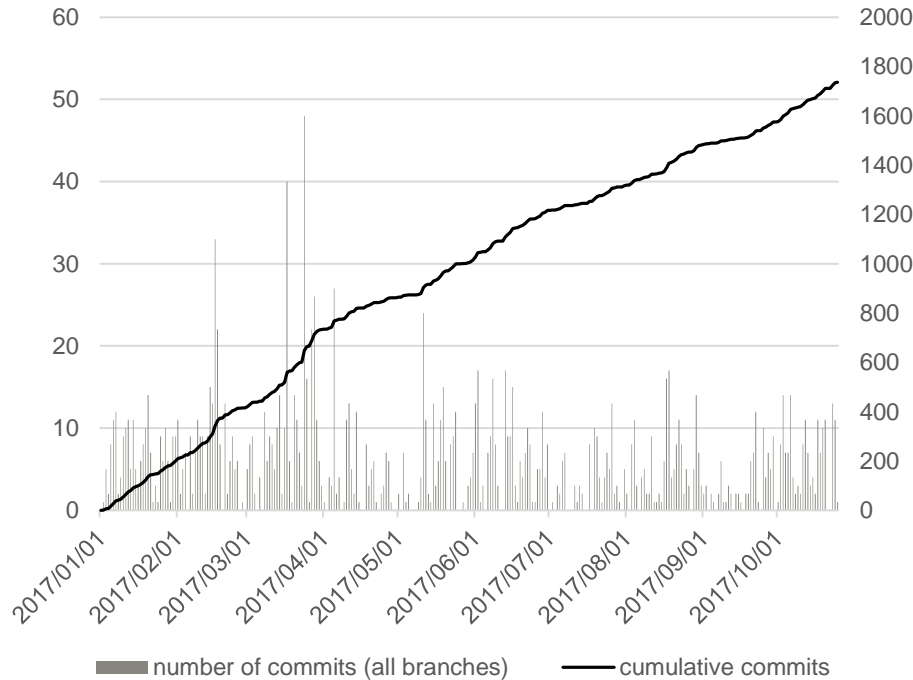
Development Process



- How is OAI growing up??

OAI Commit History

■ openair5g (UE/eNB) – all branches



Development is ongoing at the same pace

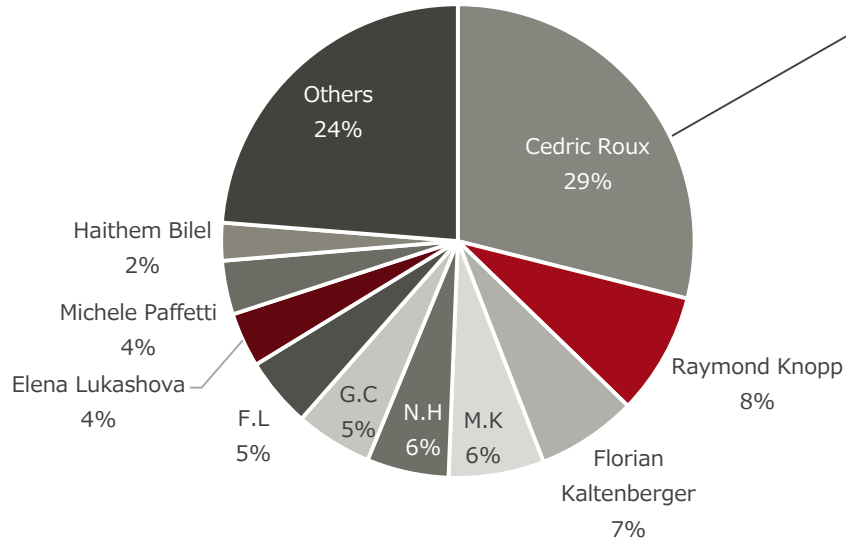
The pace of development is being maintained for the last year.

2017 Jan	199
2017 Feb	216
2017 Mar	319
2017 Apr	128
2017 May	153
2017 Jun	202
2017 Jul	100
2017 Aug	165
2017 Sep	93
2017 Oct	161 (until 25 th Oct)

From January 1st to October 25th, 2017

OAI Commit Ranking

■ openair5g (UE/eNB) – all branches Top 10

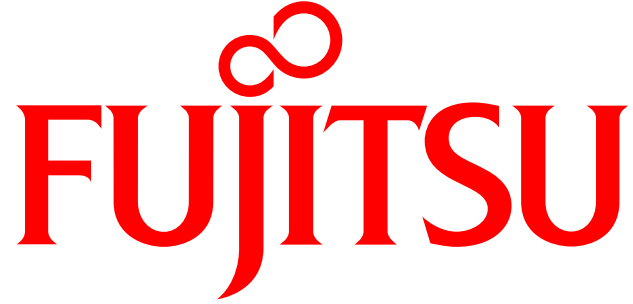


50% commit from EURECOM

More diversity preferred?

- #1 Cedric Roux
EURECOM
- #2 Raymond Knopp
EURECOM
- #3 Florian Kaltenberger
EURECOM
- #4 M.K
b<>com
- #5 N.H
Gmail
- #6 G.C
Gmail
- #7 F.L
b<>com
- #8 Elena Lukashova
EURECOM
- #9 Michele Paffetti
EURECOM (UniBo)
- #10 Haithem Bilel
TCL

From January 1st to October 25th, 2017



shaping tomorrow with you

References



(*)SArF
Kobayashi, K., Kamimura, M., Yano, K., Kato, K. and Matsuo, A.:SArF Map: Visualizing software architecture from feature and layer viewpoints, Program Comprehension (ICPC), 2013 IEEE 21st International Conference on, pp. 43-52, DOI: 10.1109/ICPC.2013.6613832 (2013).